REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Final Technical 9/30/94 through 9/30/98

4. TITLE AND SUBTITLE
A Foveated Imaging System to Reduce Transmission Bandwidth of Video Images from Remote Camera Systems

5. FUNDING NUMBERS
contract
F49620-94-C-0090

6. AUTHOR(S)

Wilson S. Geisler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

(1) OWL Displays, Inc. 3925 W. Braker Ln, Austin, TX 78712

(2) Center for Vision & Image Science Mezes Hall 330, University of Texas
Austin, TX 78712

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research 110 Duncan Avenue, Suite B115
Bolling AFB, DC 20332-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

19990119 088

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Report may be freely distributed.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

This is the final progress report for a joint project between Geisler's laboratory at the University of Texas at Austin and OWL Displays Inc., to develop a real time variable resolution (foveated) imaging system for video communications tasks such as remote control of unmanned vehicles. Although the OWL side of the project might best be described as a modest success, we believe the software and software/hardware integration performed at UT has been a big success. A general purpose real-time linkable library (for Pentium class computers running the Windows95/98/NT OS) has been developed for coding and decoding variable resolution static images and video, both in 8-bit gray scale and 24-bit color. Real-time demonstration executables using the library are currently available at our web site for this project: http://fi.cvis.psy.utexas.edu. We have tested our real time software in conjunction with MPEG (H.263) and shown that it generally produces very substantial bandwidth savings both for I frames and P frames. We have also developed our own real time image compression library which includes fast motion compensation, fast pyramid coding, fast zero-tree coding and arithmetic coding. The foveated imaging software has been successfully interfaced and tested with a 512x512 8-bit b/w camera and two separate eye trackers: an ASL desktop heads-free eye tracker and the OWL/ASL V8 helmet mounted eye tracker.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	SAR

A Foveated Imaging System to Reduce Transmission Bandwidth of Video Images from Remote Camera Systems

Final Technical Report

(prepared by W.S. Geisler)

AFOSR F49620-94-C-0090

Phase I: September 30, 1994 to March 31, 1995
Phase II: September 30, 1996 to September 30, 1998

Investigators:

W. S. Geisler
Department of Psychology &
Center for Vision and Image Sciences
Mezes 330
University of Texas at Austin
Austin TX 78712

H. L. Webb
OWL Displays Inc.
3925 W. Braker Lane
Austin TX, 78759

Submitted December 12, 1998

Introduction

This is the final progress report for a joint project between Geisler's laboratory at University of Texas at Austin and OWL Displays Inc., to develop a real time variable resolution (foveated) imaging system for video communications tasks such as remote control of unmanned vehicles.

The original objectives in Phase I were to develop fast and efficient software for encoding foveated images (to be done at UT), and to design and construct variable resolution display hardware for displaying the encoded foveated images (to be done at OWL). During Phase I a software encoder/decoder and a small test display were developed. However, the software was so efficient that conventional displays proved to be adequate for moderate sized images. Thus, in Phase II it was decided that OWL would focus only on developing a very high resolution display, while Geisler's lab would focus on improving the software encoder/decoder and on integrating it with a heads-free eye tracker. However, during Phase II a complete redesign of the software led to further increases in encoding/decoding speed and elimination of display artifacts that had existed in the earlier version on conventional displays. Thus, it became apparent by the second year of Phase II that the display technology that OWL could provide was not going to significantly enhance system performance. At that point OWL (with AFOSR approval) turned their attention to eye tracking technology to be used in conjunction with the foveated imaging software. OWL in cooperation with ASL Inc. succeeded in integrating a small eye tracker into a helmet mounted display (the Virtual Research Systems Inc. V8 helmet). This display system has been integrated and tested with the foveated imaging software. During the final year of Phase II financial difficulties overtook OWL; they filed for bankruptcy and are now out of business.

Although the OWL side of the project might best be described as a modest success, we believe the software and software/hardware integration performed at UT has been a big success.

Original Objectives

The specific original Phase I objectives were as follows:

1. Develop algorithms for foveated vision.

2. Implement and test algorithms on hardware platforms.

3. Interface camera, eye tracker and CRT hardware platform.

4. Develop a low cost monochrome screen for the foveated system based on OWL's technology

5. Develop lattice application specific integrated circuit (ASIC) interface.

6. Interface camera, eye tracker and OWL's display platform

7. Conduct a final comparison and assessment of achievable bandwidth reduction at given levels of display quality.

The specific original Phase II objectives were as follows:

1. Completion of a one-camera image acquisition system.

2. Reduction or elimination of the apparent motion effects seen in the Phase I FIS.

3. Construction of a higher resolution OWL display using a pulse driver.

4. Implementation of a heads-free eye tracking system (i.e. one that allows free head movements) and evaluation of its robustness.

5. Evaluation of human performance while using the system (for both conventional and OWL screen technologies), for a number of different image types and tasks, including visual search, text reading, and navigation through the environment.

6. Design and implementation of a data compression scheme to work in conjunction with the FIS to further reduce bandwidth requirements.

7. Evaluation of the Foveated Imaging System's transmission properties.

8. Evaluation of alternate hardware/software implementations.

Status of effort

Phase I Objectives 1, 2, 4 and 7 were completed. Objective 3 was completed except for interfacing of the camera which was accomplished during Phase II. Objectives 5 and 6 were not completed because of the change in OWL's objectives described in the Introduction. Phase II objectives 1, 2, 4, 6, 7 and 8 were all completed. Objective 3 was not completed because of another change in OWL's objectives described in the Introduction. Objective 5 was not completed in a rigorous fashion. Some behavioral testing, demonstrating the advantages of foveated imaging for optimizing search performance under fixed bandwidth transmission conditions, was completed in Phase I (Kortum & Geisler, 1996). However, for three reasons additional behavioral performance testing was not completed in Phase II. First, new insights led to a complete redesign of the software algorithms; these changes required considerable additional effort to implement. Second, the development of real-time conventional image compression algorithms appropriate for foveated imaging grew into a much larger effort than initially anticipated. Third, it was decided that higher priority should be placed upon demonstrating the compatibility of foveated imaging with existing image compression standards such as MPEG.

Work on the foveated imaging system and software libraries continues with some limited funding from UT. We plan to maintain and expand the software library for some years to maximize the chance for applications to develop.

Accomplishments/New Findings

The major accomplishments and findings are described here with an emphasis on work carried out at UT in Geisler's lab.

(1) A general purpose real-time C++ library has been developed for coding and decoding variable resolution static images and video, both in 8-bit gray scale and 24-bit color. This linkable library (for Pentium class computers running the Windows95/98/NT OS) will soon be made generally available for development and testing of applications involving variable resolution displays. We anticipate having the library available with documentation within a couple of months. Real-time demonstration executables using the library are currently available at our web site: http://fi.cvis.psy.utexas.edu.

The details of the algorithms are described in Geisler & Perry (1998) and in documentation available at the web site. In brief, a low-pass pyramid of 5-6 levels is first created. Each successive level of the pyramid is a copy of the image from the previous level, but at 1/2 the resolution in each direction (1/4 the number of pixels). Based upon the foveation points (i.e., the locations of highest resolution) subsets of pixels from each level of the pyramid are selected to create a series of smaller images. Typically, all of these smaller images together contain only a fraction of the total pixels in the original image. The subimages can then be processed in any fashion, just like normal images. For example, they can be MPEG coded and transmitted to a remote site and then MPEG decoded. The processed small images are then decoded, interpolated and smoothed by our software to obtained a displayable foveated image.

(2) We have tested our real time software in conjunction with MPEG (H.263) and shown that it generally produces very substantial bandwidth savings both for I frames and P frames. The results are described in documentation available at the web site. The results will appear in an invited SID paper (Geisler & Perry, 1999). As implied above, our foveation software was explicitly developed for pre- and post-processing, and hence it is generally compatible with a wide range of image processing hardware.

(3) We have also developed our own real time image compression software which includes fast motion estimation/compensation, fast pyramid coding, fast zero-tree coding and arithmetic coding. These are conventional state-of-the-art components of the best current image compression algorithms. Our contribution was to develop versions with very good real time performance. We have shown that our foveation coding increases the speed of subsequent processing because of the great reduction in the number of pixels that must be processed. With our software both variable resolution coding and more conventional image compression can be accomplished in real time for moderate sized images, without any special purpose hardware. A real-time C++ library of these routines will also be made generally

available for development and testing at or near the same time the foveated imaging library is made available.

(4) A foveated imaging web site has been created which contains much information about foveated imaging, including downloadable real-time demonstrations. This web site has received many hundreds of hits (from unique addresses) since its inception in September, 1998. Feedback about the demonstrations has been very positive. A number of laboratories have indicted that they plan to make use of the libraries as soon as they are available. On the basis of the web site and demonstrations, we will be giving an invited presentation on variable resolution displays at the Society for Information Display meeting in San Jose next summer. The downloadable demonstrations now include real-time foveation of static 8-bit color images and 24-bit color movies, where the user can control the foveation point with a mouse. In addition, there is a real-time simulation of fixed bandwidth telecommunications showing how foveation can dramatically increase frame rate without sacrificing field of view.

(5) We have interfaced the current foveated imaging software with a 512x512 8-bit b/w camera and two separate eye trackers: an ASL desktop heads-free eye tracker and the OWL/ASL V8 helmet mounted eye tracker. The software library contains appropriate routines for communicating with the ASL eye trackers. For most observers who have tried these eye-tracking systems the foveated imaging works quite well. However, we have found that for a small percentage of observers the eye trackers are not able to track over a wide range of gaze angles. We are planning a more systematic study of eye tracking with foveated imaging, however, we also believe that there are a number of applications where simpler pointing devices will be adequate.

(6) Two papers have been published on foveated imaging, one describing the Phase I work and one describing some of the Phase II work. A short paper will be published in the SID proceedings describing some of the most recent work.

(7) The University of Texas has filed for a patent on the foveated imaging algorithms.

(8) We have made progress in demonstrating the value of our variable resolution software in several applications including telecommunications, image retrieval and 3D simulation. For example, we are currently working on a video telecommunications test bed (with University equipment funds). Also, we have begun creating a demonstration of how foveated imaging can be used to speed 3D graphics simulation and rendering in the OpenGL environment.

Personnel Supported

Principal Investigator

Wilson S. Geisler

Technical Support Personnel

Carl Creeger
Larry Stern

Administrative Support Personnel

Christine Fry

Direct Research Personnel

Viral Kadakia
Phil Kortum
Jeffrey Perry
Joshua Siegel

Publications

Kortum, P.T. and Geisler, W.S. (1996) Implementation of a foveated image-coding system for bandwidth reduction of video images. In B. Rogowitz and J. Allebach (Eds.) *Human Vision and Electronic Imaging. SPIE Proceedings*, 2657, 350-360.

Siegel, J. (1996) Coding of foveated images. Masters Thesis.

Geisler, W.S. and Perry, J.S. (1998) A real-time foveated multi-resolution system for low-bandwidth video communication In: B. Rogowitz and T. Pappas (Eds.), *Human Vision and Electronic Imaging, SPIE Proceedings* 3299, 294-305.

Geisler, W.S. and Perry, J.S. (1999) Variable resolution displays for visual communication and simulation. Society for Information Display (SID), San Jose, June 1999.

Geisler, W.S. and Perry, J.S. website: http://fi.cvis.psy.utexas.edu.

Inventions/patents

Foveated Image Coding System and Method for Image Bandwidth Reduction. Pat App. No. 08/997,109

Implementation of a foveated image coding system for image bandwidth reduction

Philip Kortum and Wilson Geisler

University of Texas Center for Vision and Image Sciences. Austin, Texas 78712

ABSTRACT

We have developed a preliminary version of a foveated imaging system, implemented on a general purpose computer, which greatly reduces the transmission bandwidth of images. The system is based on the fact that the spatial resolution of the human eye is space variant, decreasing with increasing eccentricity from the point of gaze. By taking advantage of this fact, it is possible to create an image that is almost perceptually indistinguishable from a constant resolution image, but requires substantially less information to code it. This is accomplished by degrading the resolution of the image so that it matches the space-variant degradation in the resolution of the human eye. Eye movements are recorded so that the high resolution region of the image can be kept aligned with the high resolution region of the human visual system. This system has demonstrated that significant reductions in bandwidth can be achieved while still maintaining access to high detail at any point in an image. The system has been tested using 256x256 8 bit gray scale images with 20° fields-of-view and eye-movement update rates of 30 Hz (display refresh was 60 Hz). Users of the system have reported minimal perceptual artifacts at bandwidth reductions of up to 94.7% (18.8 times reduction)

KEYWORDS: foveation, field-of-view, gaze contingent, area-of-interest, eye movements, image compression

1.0 INTRODUCTION

The human visual system functions as a unique space-variant sensor system, providing detailed information only at the point of gaze, coding progressively less information farther from this point. This implementation is an efficient way for the visual system to perform its task with limited resources; processing power can be devoted to the area of interest and fewer sensors (i.e. ganglion cells and photoreceptors) are required in the sensor array (the eye). Remarkably, our perception does *not* reflect this scheme. We perceive the world as a single high resolution image, moving our eyes to regions of interest, rarely noticing the fact that we have severely degraded resolution in our peripheral visual field.

The same constraints that make this space-variant resolution coding scheme attractive for the human visual system also make it attractive for image compression. The goal in real-time image compression is analogous to that of the human visual system; utilization of limited resources, in this case transmission bandwidth and computer processing power, in an optimum fashion. Transmitted images typically have a constant resolution structure across the whole image. This means that high resolution information must be sent for the entire image, even though the human visual system will use that high resolution information only at the current point of interest. By matching the information content of the image to the information processing capabilities of the human visual system, significant reductions in bandwidth can be realized, provided the point-of-gaze of the eye is known.

Recently, there has been substantial interest in foveated displays. The US Department of Defense has studied and used so-called "area-of-interest" (AOI) displays in flight simulators. These foveation schemes typically consist of only 2 or 3 resolution areas (rather than continuous resolution degradation) and the center area of high resolution, the AOI, is often quite large, usually between 18° and 40° (see, for example Howard, 1989, Warner, Sefoss and Hubbard, 1993). Other researchers have investigated continuous variable resolution methods using log polar mapping (Weiman, 1990, Juday and Fisher, 1989, Benderson, Wallace and Schwartz, 1992). Log polar mapping is particularly advantageous when rotation and zoom invariance are required, but their implementations have necessitated special purpose hardware for real-time operation. We have developed a preliminary version of a system that accomplishes real-time foveated image compression and display using a square symmetric Cartesian resolution structure, implemented on a general purpose computer processor.

2.0 SYSTEM OPERATION

This Cartesian coordinate based Foveated Imaging System (FIS) has been implemented in C (Portland Group PGCC) for execution on an ALACRON i860 processor. This platform has been used merely as a testbed; preliminary tests have shown that a 7 to 10 fold *increase* in performance can be achieved when implemented on a Pentium 90 MHz processor. Figure 1 illustrates the general function of the FIS. The system is initialized and the user is queried for a number of required parameters (half resolution constant, desired visual field of the display, eye movement update threshold). Using these parameters, a space variant arrangement of SuperPixels (referred to as the ResolutionGrid) is then calculated and stored for display. A SuperPixel is a collection of screen pixels (where a screen pixel is defined as ScreenPixelSize (degrees) = 60 x display size (degrees) / image size in pixels) that have been assigned the same gray level value. The user is then prompted through an eye tracking calibration procedure in order to account for variations in head position at setup. The system then enters a closed loop mode, in which eye position is determined and compared with the last known eye position. If the eye has not moved more than some predetermined amount (specified during initialization), the pixel averaging subroutine is executed and eye position is checked again. However, if the eye has moved more than this threshold amount, then a new eye fixation location is calculated and the pixel averaging subroutine is executed, creating the gray levels for each of the SuperPixels in the ResolutionGrid. These SuperPixels are then sent to the display device, at which time eye position is checked again.

Only the closed loop portion of the program is required to run in real time. Initialization, calibration and calculation of the space-variant resolution grid take place prior to image display. However, because of the simplicity of the resolution grid structure (which is described in greater detail below), it can also be re-calculated in real time, if desired.

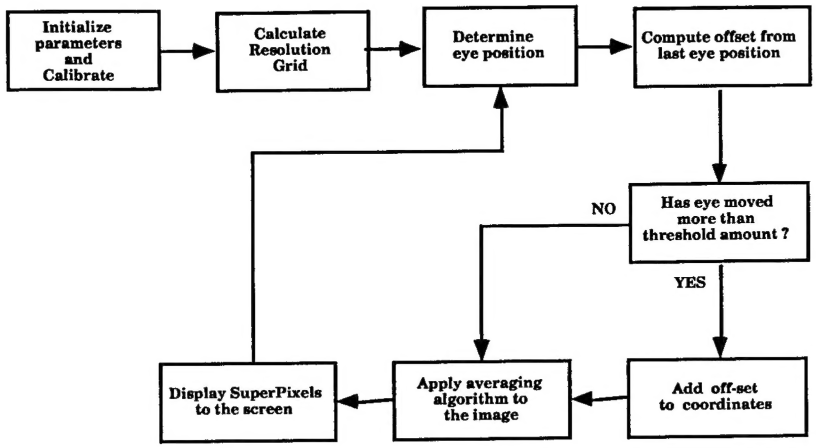


Figure 1: General flow diagram of the operation of the foveated imaging system.

2.1 Resolution fall-off calculations

The need for real time performance requires a resolution structure that results in high computational efficiency in algorithmic implementations. The square symmetric pattern, shown in Figure 2, is one such configuration. Because the resolution structure is specified in Cartesian coordinates, and each of the SuperPixels is square, pixel locations can be represented with a set of corner coordinates. This allows implementation of operations such as scaling and translation to occur using only addition.

Starting in the south-west corner of the north-east pixel in ring i (the pixel at location x_i, y_i), the size of a SuperPixel is calculated according to the formula,

$$W_i = \frac{W_0}{\sqrt{2}}\left(1 + \frac{\sqrt{x_i^2 + y_i^2}}{\varepsilon_2}\right) \tag{1}$$

where W_i is the size of the SuperPixels in ring i (in pixels), W_o is the size of the central foveal SuperPixel (in pixels), x_i and y_i are the distances along a diagonal from the center of the screen (in degrees), and e_2 is the half-resolution constant, expressed in degrees. This function is based on available perceptual data and is also consistent with anatomical measurements in the human retina and visual cortex (Wilson *et al*, 1990; Geisler and Banks, 1995; Wassle, *et al*, 1992). Specifically, when e_2 is between 0.8 and 1.2 the SuperPixel size is approximately proportional to the human resolution limit at each eccentricity. Thus, if W_o is less than or equal to the foveal resolution limit then the foveated image will be *indistinguishable* from the original image (with proper fixation). If W_o is greater than the foveal resolution limit then the foveated image *will* be distinguishable from the original image (note that because W_o is a proportionality constant in equation (1) the SuperPixel size will be above the resolution limit by a constant factor at all eccentricities).

Once the size of the SuperPixel in the NE corner of ring i is determined a three pronged decision tree is entered in order to calculate the size of the remaining SuperPixels in the ring. This is necessary because an integer number of SuperPixels of size W_i may not fit in the space delineated by the square symmetric ResolutionGrid. This means that, while all of the SuperPixels in a ring will have the same size in one direction (W_i), they may not have the same size in the other direction. The simplest situation (case 1) occurs when an integer number of SuperPixels of size W_i *can* be accommodated in the specified space. The other two branches of the decision tree (cases 2 and 3) essentially conduct multiple bisections, putting the smallest SuperPixels in the center of the side, increasing SuperPixel size in a symmetric fashion towards the corners, where the SuperPixels are W_i x W_i. Case 2 handles situations where only one reduced size SuperPixel is required. Case 3 handles situations in which multiple reduced size SuperPixels are required.

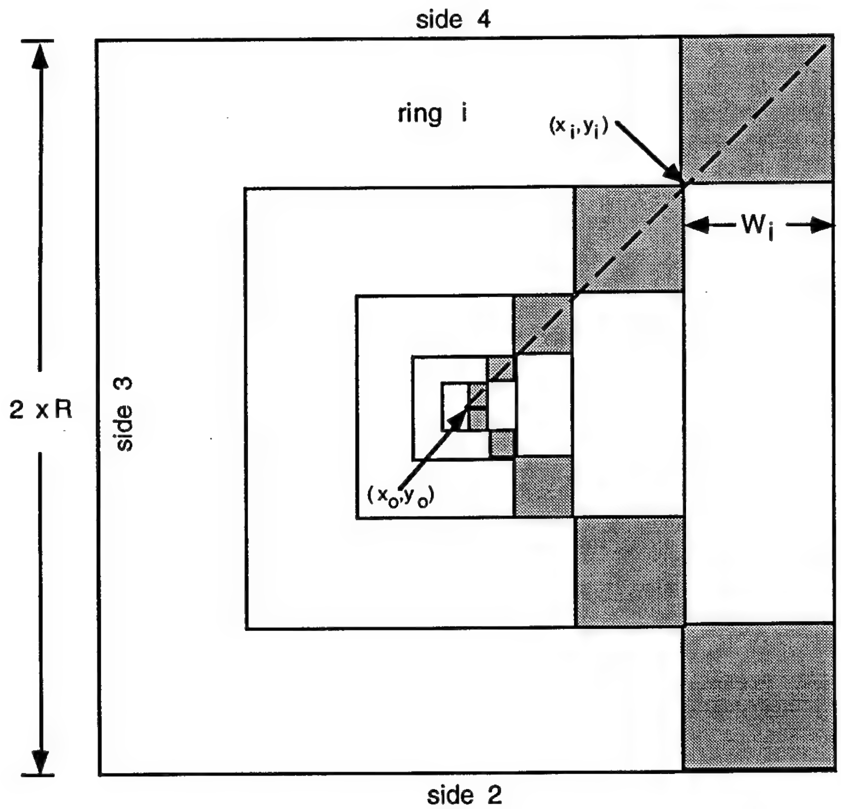


Figure 2: Foveated imaging system SuperPixel pattern arrangement, which is called the ResolutionGrid. SuperPixels of size W_i are arranged in concentric rings (i) about the origin (x_o,y_o). As described in the text, the ResolutionGrid is twice the size (4 times the area) of the viewable screen to allow for simple update as the result of an eye movement.

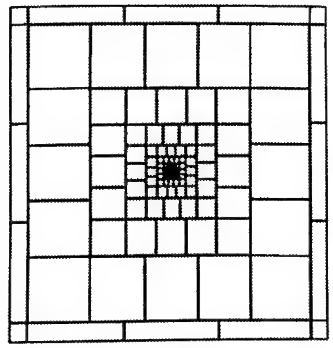
In order to increase the computational efficiency of the program, this three pronged decision calculation is only carried out for a single side (side 1, as labeled in Figure 2) of the pixel ring i. Each SuperPixel is represented by 4 numbers: the x and y locations of the lower left-hand (SW) and upper right hand (NE) corners. Because of the square symmetric SuperPixel pattern, the SuperPixel coordinates from the single computed side are simply moved through three 90 degree rotations to establish the coordinates of all the SuperPixels in the ring.

Upon completion of a ring, the program checks to see if the coordinates of the NE corner of the NE SuperPixel in the last calculated ring are greater than 2 times the resolution of the image (for reasons explained in detail in the description of the tracking subroutine); if not, the subroutine is run again. If so, the entire set of SuperPixel coordinates (the ResolutionGrid) is written to memory for later use. As mentioned earlier, since the ResolutionGrid is stored for later use, its calculation time does not affect the real time capability of the system.

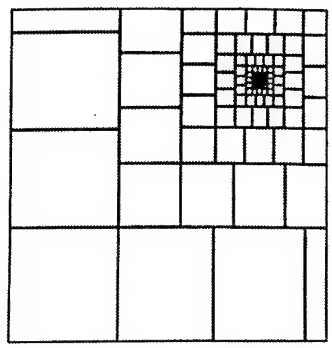
Once the structure of the ResolutionGrid has been determined, a SuperPixel averaging subroutine averages the gray levels of each of the screen pixels that fall within a SuperPixel boundary and assigns the resulting average gray level to that SuperPixel. If the entire SuperPixel does not fall in the bounds of the display device (recall that the ResolutionGrid is twice the size of the viewable image) then the average includes only displayed pixels. Because the SuperPixel averaging subroutine takes place in real time, computational efficiency is important: therefore, SuperPixels of width 1 are excluded from the averaging subroutine, and their gray levels are passed directly to the display. Once the average gray level is determined for each SuperPixel, its value is added to the ResolutionGrid, which is then passed to the screen for display.

2.2 Gaze tracking

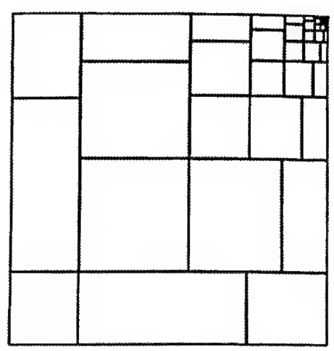
After the initial SuperPixel gray level assignment, the program enters a loop in which the position of the eye is measured and compared against the last measured eye position. If the eye has not moved more than the threshold amount (which is specified during the parameter initialization subroutine), the SuperPixel averaging subroutine is called, and the resulting averaged SuperPixels are displayed. This insures that, even with steady fixation, changes in the image (i.e. motion of an object or the video camera) will be reflected in the display. The subroutine then loops back and checks the position of the eye again . If, however, the eye has moved more than the threshold amount, several things happen. First, the amount of the movement (expressed in terms of pixels) is added to the each of the ResolutionGrid coordinates. The result is a change in the position of the high resolution region. Figure 3 shows an example of how the portion of the ResolutionGrid that is displayed changes as fixation changes; here a subject begins with center fixation, and moves his eyes towards the northeast corner of the display device. When this happens, the amount of his eye movements (in the x and y direction) are added to the current ResolutionGrid coordinates, causing the foveated region to offset the same amount. The result is an image that has highest resolution at the point of gaze. Initially calculating a ResolutionGrid that is twice the size of the viewable area (as previously shown in Figure 2) allows us to use this computationally efficient offset method to track eye position and update the display *without* having to recompute the ResolutionGrid each time the eye moves. Figure 4 illustrates how the offset works; adding the eye movement offset to the current ResolutionGrid coordinates is essentially the same as moving the ResolutionGrid to a position that coincides with current fixation location, while keeping the viewable screen in a fixed location. Since the ResolutionGrid is twice the size (4 times the area) of the viewable screen, recomputation of the ResolutionGrid is unnecessary because all eye positions in the viewable screen can be accounted for with a simple offset of the ResolutionGrid. Once the updated SuperPixel configuration on the viewable screen is determined, the SuperPixel averaging subroutine is called, and the resulting averaged SuperPixels are displayed.



Center Fixation



Towards NE Corner Fixation



NE Corner Fixation

Figure 3: An example of how the portion of the ResolutionGrid that is displayed changes as fixation changes; here a subject begins with center fixation, and moves his eyes towards the northeast corner of the display device. As he does this, the amount of his eye movements (in the x and y direction) are added to the current ResolutionGrid coordinates, causing the foveated region to offset the same amount. Notice how the SuperPixels increase in size as they become further from the point of fixation.

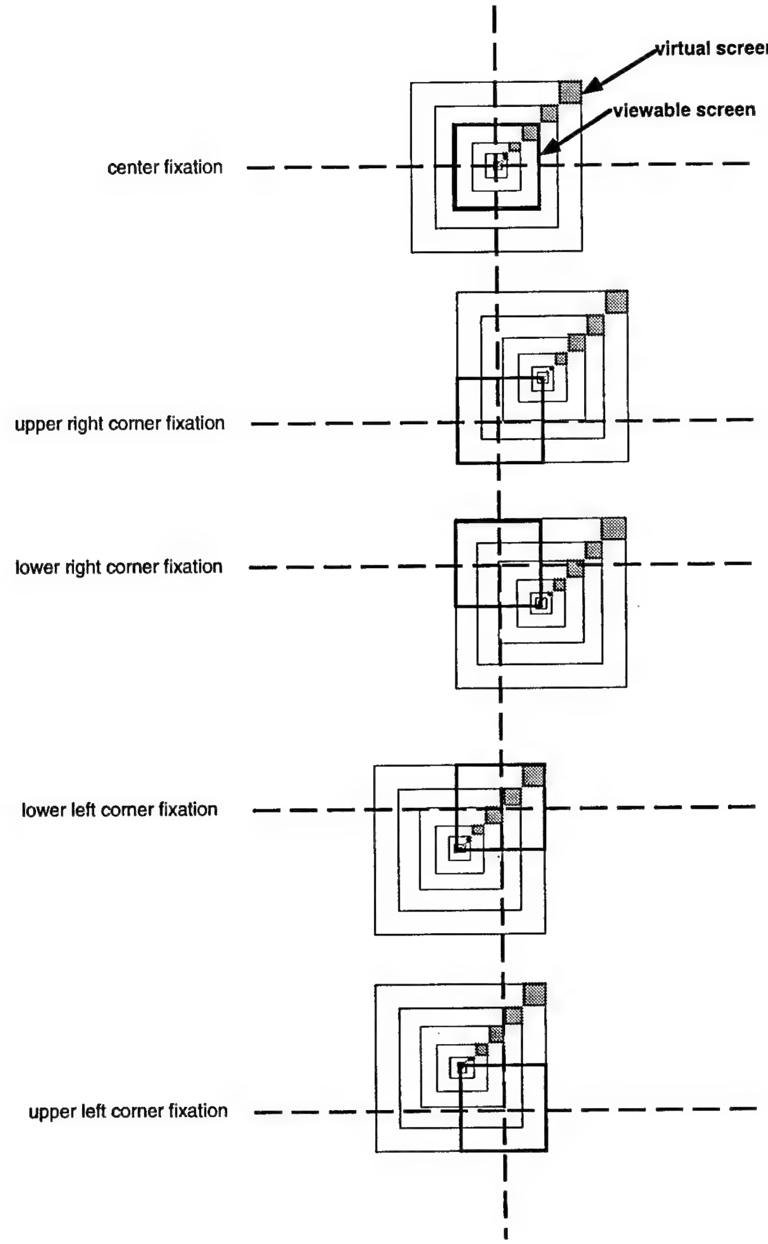


Figure 4: Calculating a ResolutionGrid that is twice the size of the viewable area allows a computationally efficient offset method to track eye position and update the display *without* having to recompute the ResolutionGrid each time the eye moves. The dark outline square is the viewable screen, with the remaining portion being the expanded ResolutionGrid, or so-called virtual screen. Adding the eye movement offset (the amount the eye has moved since the previous measurement) to the current ResolutionGrid coordinates is essentially the same as moving the ResolutionGrid to a position that coincides with current fixation location, while keeping the viewable screen in a fixed location. Since the ResolutionGrid is twice the size (4 times the area) of the viewable screen, recomputation of the Resolution Grid is unnecessary because all eye positions in the viewable screen can be accounted for with a simple offset of the ResolutionGrid. Several extreme fixation locations are illustrated here to demonstrate this effect.

3.0 FIS PERCEPTUAL EVALUATION

The Foveated Imaging System has been evaluated for perceptual performance using a conventional 21 inch VDT. Three images, as shown in Figure 5, were used in the perceptual evaluation of the FIS. The images were 256x256 8 bit images, having a 20° field of view. These images were selected to test the perception of a number of probable image types: a letter chart (evaluation of visual clarity in a reading task), a natural environment scene (evaluation of cluttered, high detail images) and a face (evaluation of telecommunication systems). User reports of the subjective quality of the display were used in the evaluation. More detailed psychophysical performance measurements are currently being undertaken.

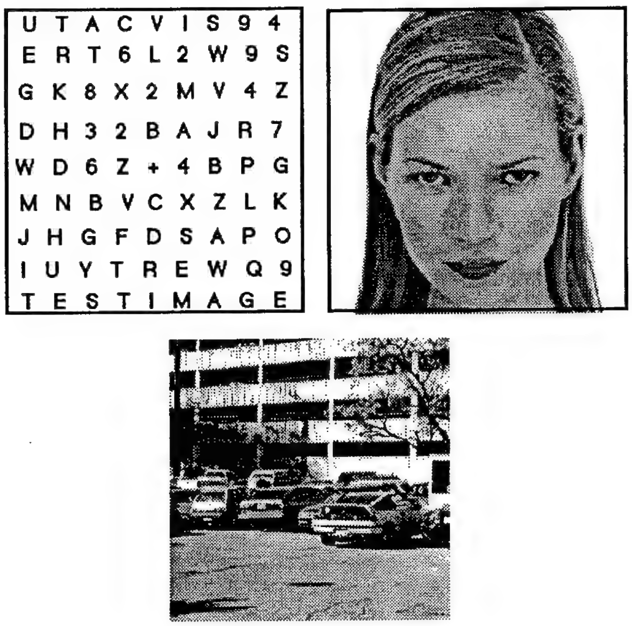


Figure 5: Three images (256x256, 8-bit gray scale) used in the perceptual evaluation of the FIS, chosen to represent the general categories of expected images.

Figure 6 shows, for example, a centrally fixated foveated image for the "letters" image with a half resolution constant (e_2) of 1°. This value of e_2 reduces the number of transmitted pixels from 65,536 to 3,488 (a factor of 18.8). All subjects reported smooth, accurate tracking and excellent overall functioning of the foveating algorithms. Upon steady fixation , most subjects noted that they were aware of the reduced resolution in the peripheral visual field, but that the effect was minimal and had only a small impact on the perceived quality of the image.

High contrast images (like the letter chart) were also reported to exhibit some reduction in the perceived contrast in the periphery, as compared to an unfoveated image. The effect was significantly reduced in the natural and face images, where contrast changes in the image are typically smoother. Without reference to the unfoveated image (i.e. without switching between the two images), few subjects were even aware of this peripheral reduction in contrast for the natural scene and face images.

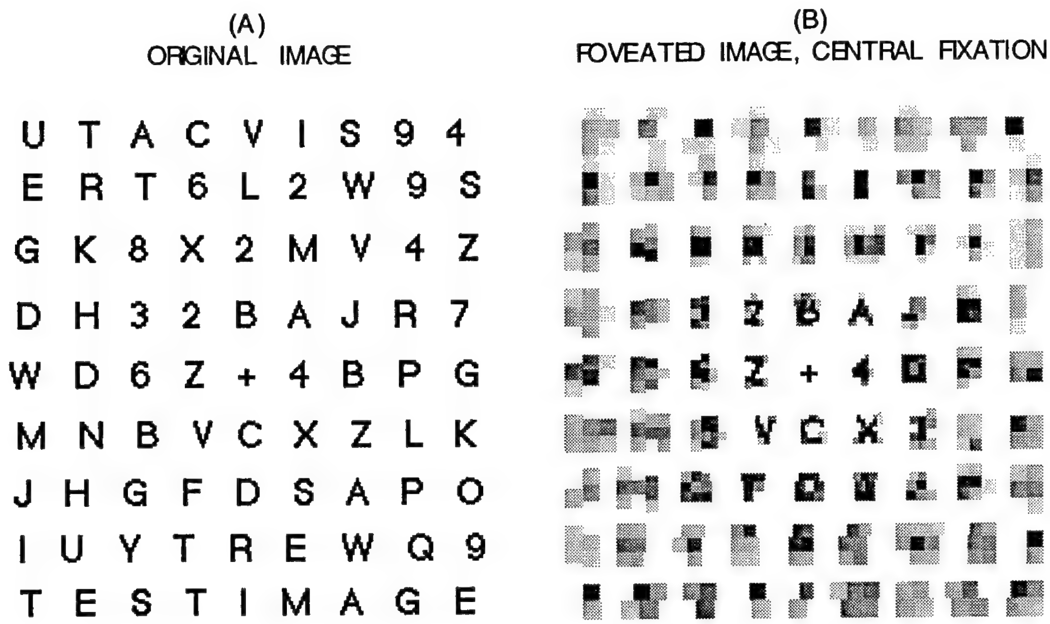
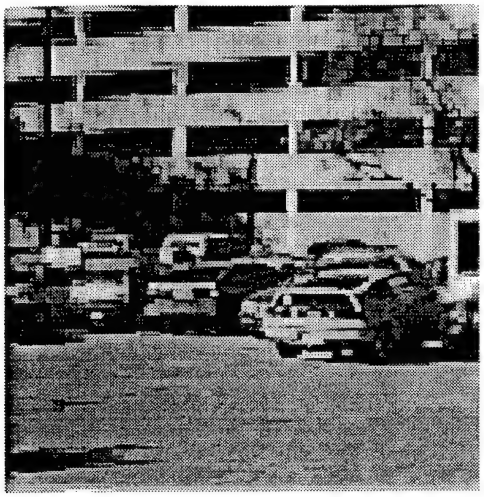


Figure 6: (A) The original 256x256 high contrast letter image. (B) The same image in a foveated viewing scheme, centrally fixated. Notice how the size of the SuperPixels grows as eccentricity increases.

Subjects also reported some apparent motion effects in the periphery (best described as 'image sparkling'), even at steady fixation. This effect was exacerbated by image updates due to small eye tremors and micro saccades, a side effect of using a high precision eye tracking system. Increasing the eye movement threshold parameter eliminated the apparent motion effects at steady fixation. However, eye movements around the image still result in apparent motion effects in the periphery. The effects are not substantial. However, for certain tasks, such as piloting a remote vehicle, the separation of apparent motion from real motion could cause some difficulties. Some of these artifacts can be minimized through the application of post-transmission SuperPixel averaging, resulting in a blurring of the image outside the unaltered point-of-gaze. We have begun to implement and evaluate a number of these post-transmission averaging methods, and preliminary results indicate that there is an improvement in the perceptual quality of the image following this type of image operation.

All of these perceptual artifacts are undoubtedly due, in large part, to the fact that a 256x256 image results in screen pixels that are larger than the resolution limit of the eye. In other words, the smallest possible SuperPixel (1 screen pixel) is resolvable in the center of the fovea. SuperPixel size grows with eccentricity and hence remains above the human resolution limit (by a fixed proportion) at each eccentricity. When the SuperPixels are above the resolution limit they will produce reduced contrast, motion aliasing and visible SuperPixel edges. Increasing the resolution of the image so the screen pixels are at the resolution limit in the fovea (e.g., a 1024 x 1024 image viewed at 20°) results in foveated images that are virtually indistinguishable from the original (see Figure 7), while still reducing the number of transmitted image pixels by a factor of 18.8.



(A)

1024x1024 Foveated Image

(B)

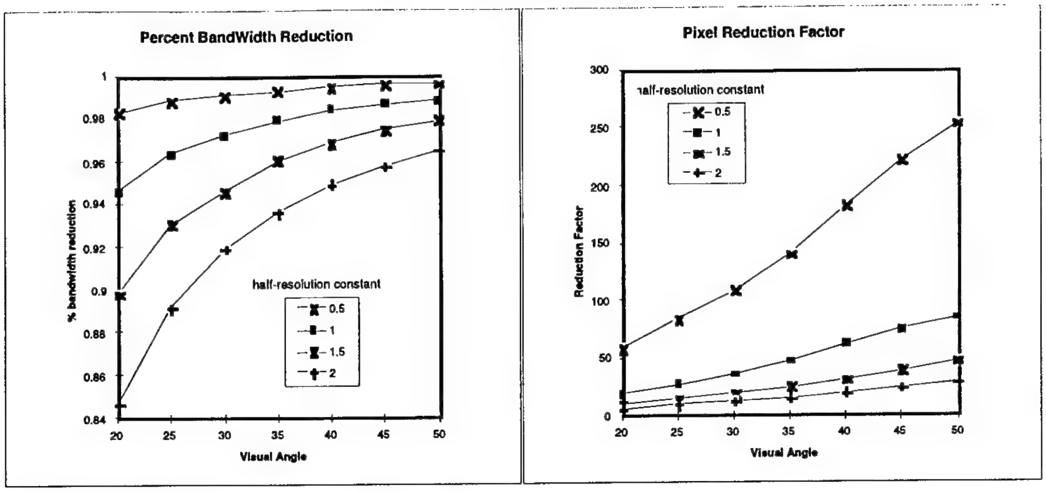
256X256 Foveated Image

Figure 7: (A) A 1024x1024 foveated image fixated on the license plate of the Honda CRX. Because the center pixel size is below the resolution limit of the visual system (for this size image), the resulting degradations due to foveation are imperceptible with appropriate fixation (notice the blockiness of the tree for verification that it is, indeed, a foveated image). (B) a 256x256 foveated image with the same fixation, for comparison.

4.0 BANDWIDTH REDUCTION

Use of the Foveated Imaging System has demonstrated that significant bandwidth reduction can be achieved, while still maintaining access to high detail at any point of a given image. Using a 20° field-of-view, a half resolution constant (e_2) of 1° and a foveal SuperPixel size of 1, we are able to achieve bandwidth reductions of 94.7% (18.8 times reduction) for 256x256 8-bit gray scale images at eye-movement update rates of up to 20 Hz (the refresh rate of the display was 60 Hz.). Increasing the field-of-view or decreasing the half-resolution constant will result in greater bandwidth savings. For example, for a 50° field-of-view, bandwidth is reduced by a factor of 96.4. A selected sampling of bandwidth reductions for different half-resolution constants and fields-of-view is shown in Figure 8.

It is important to note that other image compression schemes, such as DPCM or run length coding, for example, can be easily used in conjunction with the foveated imaging system. The effects of the supplementary compression are multiplicative; a factor of 4 compression on the foveated image described above would yield an overall compression of 75.2. This suggests that extremely high compression ratios are attainable using computationally efficient coding methods.



(a) (b)

Figure 8: Bandwidth reduction for 4 half resolution constants (e_2) as a function of the visual angle of the display device, expressed as (a) percent bandwidth reduction compared to an unfoveated display and (b) a reduction factor (i.e. the factor by which the number of pixels is reduced).

5.0 CONCLUSIONS

The Foveated Imaging System has demonstrated that real-time image compression can be achieved on general purpose computer processors, with no perceptual loss at the point of gaze and minimal perceptual anomalies in the peripheral visual field. Use of post transmission filtering and expansion to higher resolution images (where the screen pixels are near the resolution limit o the human visual system) should result in highly compressed images that are perceptually indistinguishable from an the constant resolution image from which they were formed.

6.0 REFERENCES

Benderson, B.B., Wallace, R.S. and Schwartz, E.L. (1994) A miniature pan-tilt actuator: the spherical pointing motor. IEEE Transactions Robotics and Automation. Vol. 10, 298-308.

Geisler, W.S. and Banks, M.S. (1995) Visual Performance. In Bass, M. (Ed.) Handbook of Optics Volume 1: Fundamentals, Techniques and Design, 2nd Edition. New York: Mcgraw-Hill.

Howard, C.M. (1989) Display Characteristics of Example Light-Valve Projectors. AFHRL-TP-88-44. Operations Training Division, Air Force Human Resources Laboratory, Williams AFB, AZ.

Juday, R.D. and Fisher, T.E. (1989) Geometric Transformations for video compression and human teleoperator display. SPIE Proceedings: Optical Pattern Recognition, Vol. 1053, 116-123.

Warner, H.D., Serfoss, G.L. and Hubbard, D.C. (1993) Effects of Area-of-Interest Display Characteristics on Visual Search Performance and Head Movements in Simulated Low-Level Flight. AL-TR-1993-0023. Armstrong Laboratory, Human Resources Directorate, Aircrew Training Division, Williams AFB, AZ.

Wassel, H., Grünert, U., Röhrenbeck, J., and Boycott, B.B. (1990) Retinal ganglion cell density and cortical magnification factor in the primate. Vision Research, 30, 1897-1911.

Weiman, C.F.R. (1990) Video Compression Via Log Polar Mapping. SPIE Proceedings : Real Time Image Processing II, Vol. 1295, 266-277.

Wilson, H.R., Levi. D., Maffei, L., Rovamo, J. and Devalois, R. (1990). The Perception of Form: Retina to Striate Cortex. In L.S. & J.S. Werner (Eds.), Visual Perception: The Neurophysiological Foundations (pp 232-272). San Diego: Academic Press.

PROCEEDINGS OF SPIE REPRINT

SPIE—The International Society for Optical Engineering

Reprinted from

Human Vision and Electronic Imaging III

26–29 January 1998
San Jose, California

SPIE
P
PROCEEDINGS
SERIES

Volume 3299

A real-time foveated multiresolution system for low-bandwidth video communication

Wilson S. Geisler and Jeffrey S. Perry

Center for Vision and Image Sciences, University of Texas, Austin, TX 78712

ABSTRACT

Foveated imaging exploits the fact that the spatial resolution of the human visual system decreases dramatically away from the point of gaze. Because of this fact, large bandwidth savings are obtained by matching the resolution of the transmitted image to the fall-off in resolution of the human visual system. We have developed a foveated multiresolution pyramid (FMP) video coder/decoder which runs in real-time on a general purpose computer (i.e., a Pentium with the Windows 95/NT OS). The current system uses a foveated multiresolution pyramid to code each image into 5 or 6 regions of varying resolution. The user-controlled foveation point is obtained from a pointing device (e.g., a mouse or an eyetracker). Spatial edge artifacts between the regions created by the foveation are eliminated by raised-cosine blending across levels of the pyramid, and by "foveation point interpolation" within levels of the pyramid. Each level of the pyramid is then motion compensated, multiresolution pyramid coded, and thresholded/quantized based upon human contrast sensitivity as a function of spatial frequency and retinal eccentricity. The final lossless coding includes zero-tree coding. Optimal use of foveated imaging requires eye tracking; however, there are many useful applications which do not require eye tracking.

Key words: foveation, foveated imaging, multiresolution pyramid, video, motion compensation, zero-tree coding, human vision, eye tracking, video compression

1. INTRODUCTION

When a communication involves transmitting information that will ultimately be consumed by human observers, it is often possible to reduce transmission bandwidth requirements by exploiting the limitations of human perception. Specifically, bandwidth requirements can be lowered by transmitting only that information which the human sensory systems are capable of encoding and using. Four major human perceptual limitations have been exploited in the development of real-time video communication systems. First, the temporal contrast sensitivity of the human visual system declines at high frequencies creating a temporal resolution cutoff of approximately 60 Hz. Second, the spatial contrast sensitivity of the human visual system declines at high frequencies creating a spatial resolution cutoff of approximately 50 cycles per degree (cpd). Third, chromatic information is encoded in the human visual system by only three broad-band photoreceptors, with peak sensitivities at 440, 540 and 570 nm. Fourth, the chromatic spatial resolution of the human visual system is lower than the luminance spatial resolution by a factor of approximately two.

There is, however, a fifth major human perceptual limitation that has not been fully exploited. Namely, the spatial resolution of the human visual system declines dramatically and smoothly away from the point of fixation (direction of gaze) such that the resolution cutoff is reduced at a factor of two at 2.5 degrees from the point of fixation, and by a factor of ten at 20 degrees. In principle, large savings in transmission bandwidth can be obtained by matching the spatial resolution of the transmitted images to the fall off in spatial resolution of the human visual system.

Acceptance of foveated imaging as a useful image compression tool has been slow to develop because perceptually lossless (or nearly lossless) systems generally require tracking the position of the eye in real time, so that the high resolution region of the display can be kept aligned with the high resolution region of the eye (the fovea). Although eye tracking is practical in some applications, it is relatively expensive and complicated. However, a strong case can be made for the value of foveated imaging in a number of situations where eyetracking is not practical (see section **11**).

There have been attempts to use foveated imaging in low-bandwidth video communications. Early real-time systems used special purpose hardware, and created foveated images by increasing pixel-element size as a function of angular distance (eccentricity) from the point of fixation.[1-5] More recently, Silsbee, Bovik & Chen[6] describe a foveated block pattern matching algorithm, which Barnett & Bovik[7] subsequently demonstrated has good real-time performance. Similarly, two

Part of the IS&T/SPIE Conference on Human Vision and Electronic Imaging III
San Jose, California • January 1998 • SPIE Vol. 3299 • 0277-786X/98/$10.00

years ago, Kortum & Geisler[8] described a real-time foveated imaging system that uses a general-purpose computer, and standard camera hardware. The system is able to foveate 8-bit, 256x256 images at around 18 frames/sec. However, all of these systems suffer from two important limitations: (1) the appearance of blocking artifacts and/or motion aliasing in the periphery with moderate degrees of foveation, and (2) the lack of a natural path for incorporating recent advances in multiresolution methods of image compression. To address these limitations, we have begun development of a real-time system for foveated imaging which is based upon multiresolution pyramid coding (see also, Chang & Yap.[9]) With multiresolution pyramid coding, an image is decomposed into a pyramid of 2D arrays of coefficients representing different spatial frequency bands. The first level of the pyramid contains the greatest number of coefficients and the highest spatial frequency band. Each successive level of the pyramid contains one fourth the number of coefficients of the previous level, and encodes the band of spatial frequencies centered at one half of the center spatial frequency of the previous level.

The foveated multiresolution pyramid (FMP) imaging system described here uses standard PCs, running Windows 95 or Windows NT, and does not require special purpose signal processing hardware.

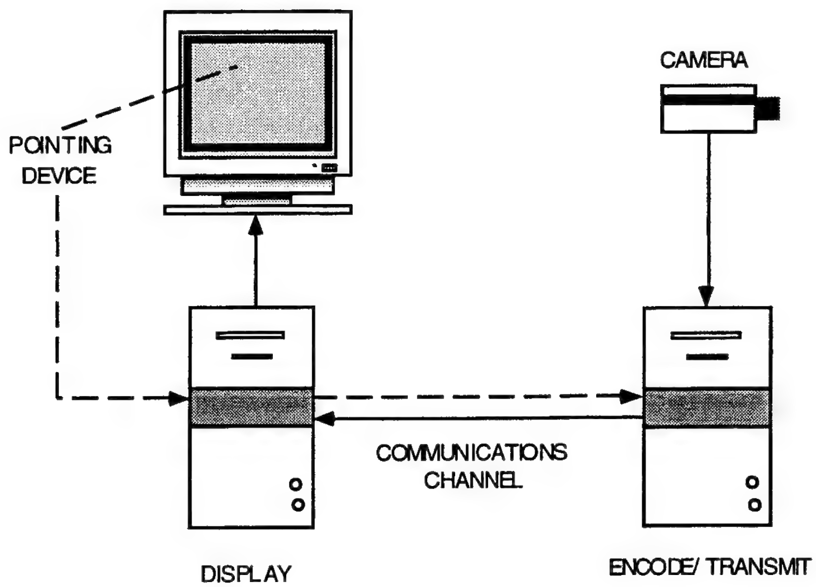


Figure 1. A foveated imaging system that is appropriate for tasks such as surveillance, teleoperation and telemedicine.

2. SYSTEM OVERVIEW

There are many potential applications of foveated imaging in real-time video communications. In some of these applications, such as surveillance, teleoperation and telemedicine, a user at one location controls the image data received from a camera at a remote location. The operation of a foveated imaging system in these applications is illustrated in Figure 1. First, the location of a foveation point is determined in real time (frame-by-frame) using some pointing device. The pointing device might be a mouse, a touch pad, or an eyetracker. The foveation point is the image location where the image will be displayed at highest resolution. Second, the coordinates of the foveation point are transmitted to the remote computer. Third, the remote computer captures a camera image. Fourth, the camera image is foveated; that is, the camera image is encoded so that the resolution of the image decreases away from the foveation point. The net result is that the degree of data compression increases with the distance from the foveation point. Fifth, the encoded image is transmitted to the local computer. Sixth, the received image is decoded and displayed on the video monitor such that the highest resolution region is centered at the foveation point. These six steps are repeated continuously in a closed loop.

A flow diagram illustrating the sequence of processing for the encoding and decoding of video image data in the FMP imaging system is given in Figure 2. Once the foveation point has been received at the remote computer, formulas based upon human psychophysical data are used to determine a foveation region for each level of the multiresolution pyramid. The foveation region is the set of pyramid elements in a level that will be further processed; no computations are done outside this region. Because spatial resolution decreases away from the fixation point, the foveation regions cover smaller fractions of the image at the lower levels of the pyramid.

An important advantage of implementing foveation in a multiresolution pyramid is that it is unnecessary to process pyramid coefficients outside the foveation region, in any given level. This makes the computation time of *every* step in the foveated codec substantially less than the computation time for a comparable non-foveated codec.

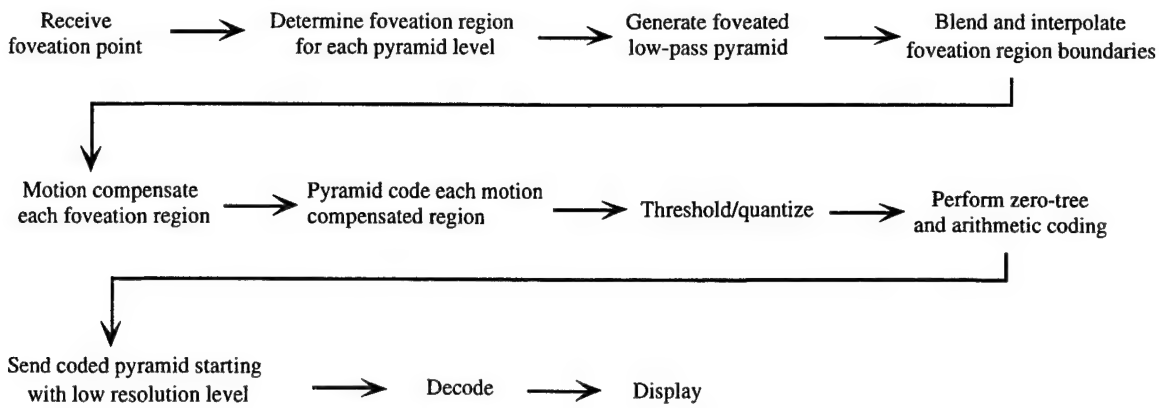


Figure 2. General flow diagram for the foveated multiresolution pyramid (FMP) imaging system.

The next step is to compute a foveated low-pass multiresolution pyramid. We use the "reduce" operation of a simple Laplacian pyramid,[10] and then select the foveated regions in each level for further processing. The next step consists of blending and foveation-point interpolation. Blending creates a smooth transition between levels of the pyramid at the boundaries created by foveation. Foveation-point interpolation incorporates the fact that a one pixel shift in the foveation point at a given level of the pyramid corresponds to fractions of a pixel shift at higher levels of the pyramid. Blending and foveation-point interpolation are important for producing smooth, artifact free foveation. The next step is to find local motion estimates for each foveation region in the pyramid, by comparing the current frame with the previous frame. We use a hierarchical block estimation method. The heirarchical motion estimation can make use of the same multiresolution pyramid used for foveation. The local motion estimates are then used to motion compensate each foveated region in the pyramid. (For a general review of motion compensation, see Tekalp.[11]) Motion compensation of each level of the pyramid is important in foveated imaging because it allows for faster processing; specifically, the compensation is only applied to the image data that will actually be transmitted. Next, each compensated foveation region is separately coded in a multiresolution pyramid. In the current version we use the Laplacian pyramid because of its excellent real-time performance; however, we expect useable real-time performace (and better compression) with a wavelet pyramid.[12, 13] The next step is to threshold and quantize the pyramid coefficients. A great deal of flexibility is available, but we have obtained good results using psychophysical measurements (contrast sensitivity data) as the basis for thresholding and quantizing as a function of both spatial frequency (level of the pyramid) and eccentricity (distance from the foveation point). The next step is lossless coding, which includes zero tree and arithmetic coding. Zero-tree coding exploits the fact that coefficients which are zero at a given level of the pyramid are likely to be superordinate to zeros in the lower levels, and thus it is often possible to code a whole "tree" of zeros with a special symbol.[14, 15] Following the lossless coding, the image data are transmitted beginning with the highest level of the pyramid (i.e., the lowest resolution data). Finally, the received data are decoded and displayed. We now describe each of these steps in more detail.

Note that two multiresolution pyramids are computed, one for foveation/motion-estimation, and another for final coding. Although this may seem inefficient, it is not. The simple, but fast, initial pyramid is sufficient for foveation and motion estimation. The foveation quickly strips away all of the image data that does not need to be processed further. The initial pyramid also allows for very fast motion compensation, which (in our experience) must occur before final pyramid coding in order to be most effective. The more complex final pyramid coding is applied to the smallest amount of data possible.

3. FOVEATED LOW-PASS MULTIRESOLUTION PYRAMID

Our method of computing the foveated low-pass pyramid is illustrated in Figure 3. The first step is to perform a "reduce" operation like that used in a Laplacian pyramid.[10] The input image (level 1) is low-pass filtered and then down-sampled by a factor of two in both directions to obtain a lower resolution image (level 2) with one quarter the number of elements. This process of low-pass filtering and down sampling is repeated to obtain a sequence of successively lower

resolution images; typically five or six resolution levels are computed, although only four are shown in Figure 3. From each of the levels we then select regions which define the amount of foveation. The inner solid squares in the upper row show the outer boundry of the foveation regions which are illustrated in the lower row. The inner dashed squares show the region in a level of the pyramid represented by the solid square in the previous level; they determine the inner boundaries of the foveation regions. In other words, the shaded regions indicate the image elements that will be processed further. In practice the inner boundaries are made a little smaller to allow blending between pyramid levels (see below). As can be inferred from this diagram, foveation can dramatically reduce both the amount of image data that must be coded and transmitted, and the total number of computations that must be performed.

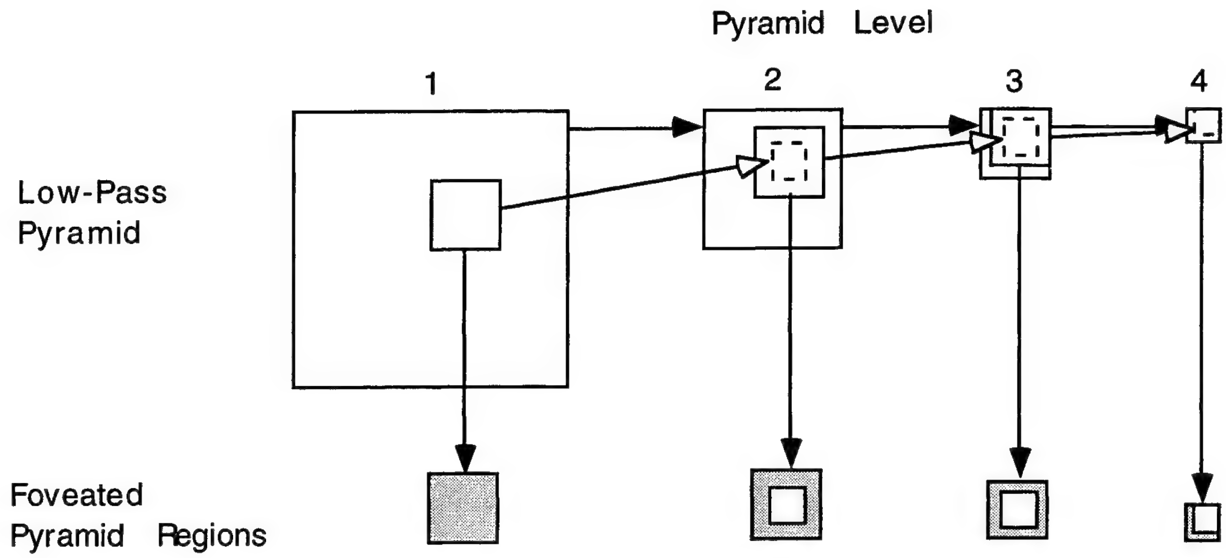


Figure 3. Schematic for the computation of a foveated low-pass multiresolution pyramid.

In the current system, the foveation regions are determined using the following contrast threshold formula, which is based upon human contrast sensitivity data measured as a function of spatial frequency and retinal eccentricity:

$$CT(f,e) = CT_0 \exp\left(\alpha f \frac{e+e_2}{e_2}\right) \tag{1}$$

where f is spatial frequency (cycles per degree), e is the retinal eccentricity (degrees), CT_0 is the minimum contrast threshold, α is the spatial frequency decay constant, and e_2 is the half-resolution eccentricity. This formula was selected because of its simplicity and because it fits published contrast sensitivity data for small, briefly presented patches of grating, which are the most relevant contrast sensitivity data for predicting detectability under naturalistic viewing conditions. The solid curves in Figure 4 show the fit of equation (1) to the contrast sensitivity data (symbols connected by dashed lines) of Robson & Graham[16]. Equation (1) also provides an adequate fit to the data of Arnow & Geisler[17] and Banks et al.[18] (see the caption to Figure 4).

Equation (1) can be used to find the critical distance from the foveation point, e_c, beyond which a given spatial frequency will be invisible (below threshold) no matter what its contrast. Specifically, the critical eccentricity can be found by setting the left side of equation (1) to 1.0 (the maximum contrast) and solving for e:

$$e_c = \frac{e_2}{\alpha f} \ln\left(\frac{1}{CT_0}\right) - e_2 \tag{2}$$

To apply equation (2), we convert into pixel units by taking into account viewing distance, and we set f to be the Nyquist frequency associated with each level of the pyramid (the highest frequency that can be reliably represented at that level). The resulting values of e_c (and the foveation point, x_0, y_0) define the foveation regions for each level of the pyramid.

Matching the foveation to the falloff in resolution of the human visual system with eccentricity makes optimal use of foveation, because it removes just that image information which cannot be resolved. However, in practice, we allow the user to control the degree of foveation by selecting the minimum contrast threshold, CT_0, and a minimum unfoveated radius, r_0. Raising CT_0 above the psychophysically measured value produces visible image degradation which is distributed across the visual field; however, there are many tasks where some distributed degradation will not reduce user performance. Similarly, there are tasks where it is important that the unfoveated region of the image not be less than some minimum size.

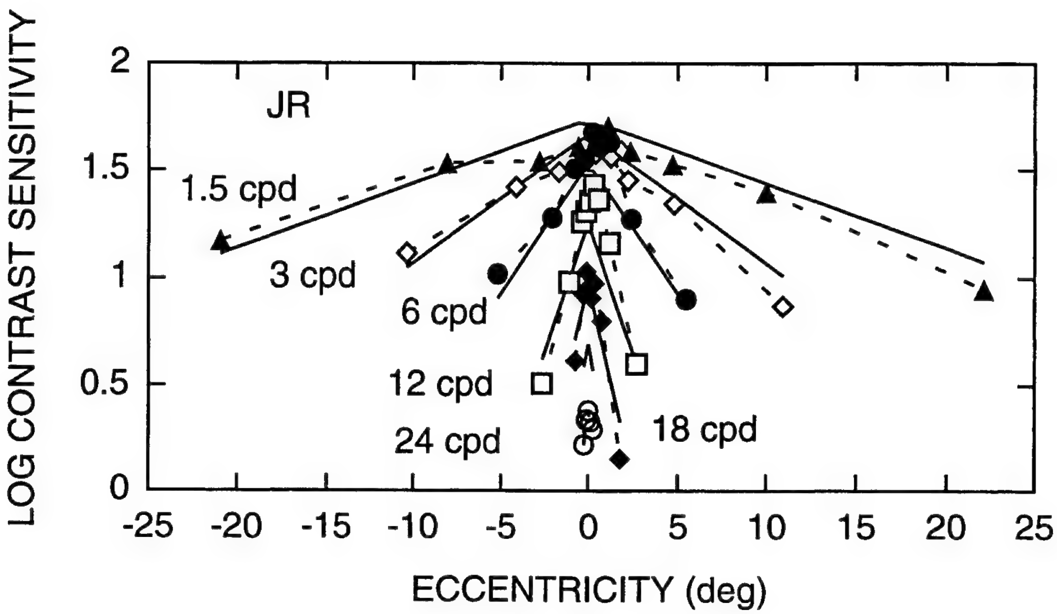


Figure 4. Contrast sensitivity (1/contrast threshold) for patches of sinusoidal grating as function of retinal eccentricity (degrees of visual angle), for a range of spatial frequencies. The symbols and connecting dashed lines are the measurements reported by Robson & Graham (1981); the solid curves are the predictions of equation (1). The best fitting parameter values (least squares fit in log units) are: $\alpha = 0.106$, $e_2 = 2.3$, $CT_0 = 1/64$. The same parameters values for α and e_2 provide a good fit to the contrast sensitivity data of Arnow & Geisler[17] with $CT_0 = 1/75$, and an adequate fit to the data of Banks et al.[18] with $CT_0 = 1/76$.

4. BLENDING AND INTERPOLATION

In foveated multiresolution pyramids there can be visible boundaries at the edges of the foveation regions, where the spatial frequency content usually changes abruptly. These foveation boundary artifacts are most visible when there is image motion or movement of the foveation point. However, they can be minimized by applying a blending function, in our case a raised cosine function, near the border of the foveation region at each level of the low-pass pyramid. Specifically, we multiply the outer edge of the foveation region by the following blending function:

$$b(x,y) = \begin{cases} 0.5\cos\left(\dfrac{\pi(e - e_c + w)}{w}\right) + 0.5 & \text{if} \quad e_c - w < e < e_c \\ 1 & \text{if} \quad e \le e_c - w \\ 0 & \text{if} \quad e \ge e_c \end{cases} \tag{3}$$

where, $e = \sqrt{(x - x_0)^2 - (y - y_0)^2}$ and w is the width of the blending region. The inner edge is multiplied by a similar function, but with the width of the blending region set to $w/2$.

Another kind of artifact can arise when the foveation point is moved. The simplest method of foveating is to set the foveation region boundary to fall at the nearest element consistent with the value of e_c given by equation (2). However, the area of the image represented by an element increases by a factor of 4 at each level of the pyramid, and thus, for example, the foveation point must move a distance of 16 pixels in the image for the foveation boundary to move 1 element in the fifth level of the pyramid. As a result, when the foveation point is moved smoothly, the boundaries of the foveation regions in the reconstructed image jump abruptly by a distance that increases as the level of the pyramid increases. As might be expected, these jumps are most apparent for the boundaries in the higher levels of the pyramid. This problem can be effectively handled by interpolation at the foveation boundaries. Let, $x_0' + \Delta x,\ y_0' + \Delta y$ be the location of the foveation point, for some level of the pyramid, expressed in units of elements. In this notation, x_0' and y_0' are integers which represent the location of a whole element (the truncated coordinates of the foveation point), and Δx and Δy are fractions between 0 and 1 which represent offsets from the whole element. Now, let x_l, y_l and x_h, y_h be the lower left and upper right corners of the foveation region assuming a foveation point exactly at x_0', y_0'. To interpolate, we obtain the slightly larger foveation region, L(x,y), defined by x_l, y_l and $x_h + 1, y_h + 1$ and then modify the region at the boundary as follows:

$$
\begin{aligned}
L(x_l, y) &\leftarrow (1-\Delta x)L(x_l, y), \\
L(x, y_l) &\leftarrow (1-\Delta y)L(x, y_l), \\
L(x_h+1, y) &\leftarrow \Delta x L(x_h+1, y), \\
L(x, y_h+1) &\leftarrow \Delta y L(x, y_h+1)
\end{aligned}
\qquad (4)
$$

This procedure produces smooth apparent motion of the foveation region.

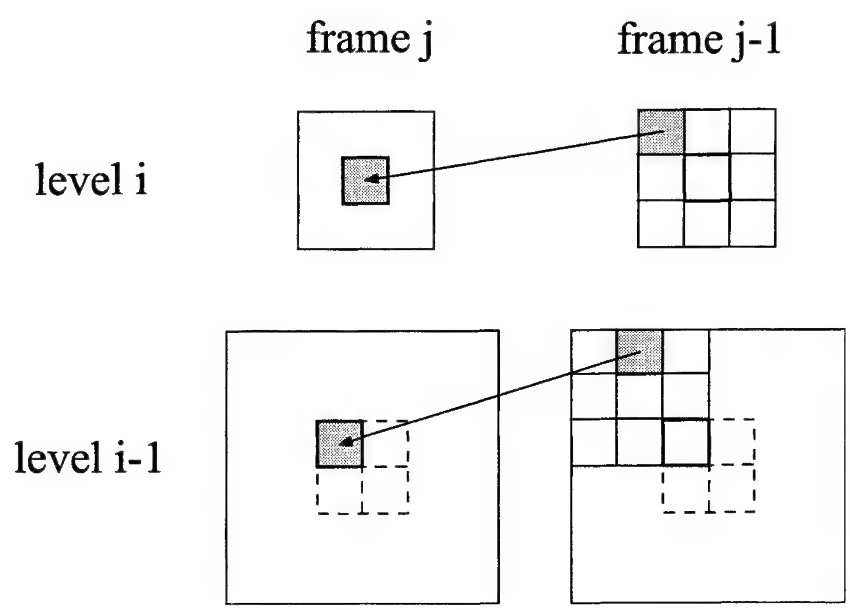


Figure 5. Hierarchical motion estimation in a multiresolution pyramid.

5. MOTION ESTIMATION AND COMPENSATION

Foveated multiresolution pyramids lend themselves readily to real-time estimation of local motion vectors from interframe comparisons. We use the hierarchical block matching method which is illustrated in Figure 5 to compute the motion vectors for each low-pass pyramid image. Specifically, we generate a low-pass multiresolution pyramid for each of the foveated subimages. These low-pass motion estimation pyramids do not have to be computed since they are contained in the previously computed foveation pyramid. The local motion for a block in the current frame (*j*) is estimated by finding the block of elements in the previous frame (*j-1*) that best matches the block in the current frame (the goodness of fit is taken to

be the sum of the absolute differences of the element gray levels). The matching procedure begins at the highest (lowest resolution) level of the motion estimation pyramid and proceeds down level-by-level to the lowest (highest resolution) level.

Figure 5 illustrates the procedure for two successive levels of the motion estimation pyramid (for simplicity we illustrate a block size of 1 although we use a block size of 8 or 16). The shaded blocks in frame *j* show the blocks that are being matched; the shaded blocks in frame *j-1* show the blocks that are the closest match to the blocks in frame *j*. As indicated by the small solid squares in frame *j-1*, nine different matches are computed for each block. In this example, the nine blocks in frame *j-1* are centered on the location corresponding to the block in frame *j*, and the upper left block provides the best match to the block in frame *j*.

A useful aspect of the pyramid representation is that matches obtained at level i provide information that can constrain the search space for matches at level *i-1*. A block at level *i* corresponds to four blocks at level i-1. For example, in Figure 5, the shaded block in frame *j* at level i corresponds to the four blocks inside the dashed square in level *i-1*. Because the best match at level *i* was in the upper left direction, that direction is the most probable for a best-match (for any one of the four blocks at level *i-1*). Thus, the search space for the shaded block in frame j of level *i-1* is given by the 9 blocks indicated by solid lines in frame *j-1*. The best match for this block is 2 blocks up and 1 block over. This example, demonstrates how hierarchical matching is able to find matches over extended regions in the image, despite the ± 1 block search space at each level.

To describe the matching process more formally, let x_i, y_i be the coordinates (in units of elements) of a given block in level *i* of frame *j* which is to be matched against blocks in frame *j-1*, and let x_i^-, y_i^- be the coordinates of the block in level *i* of frame *j-1* that best matches the block at x_i, y_i. The values of x_i^-, y_i^- can be expressed in terms of the coordinates of the starting block for the search s_i^-, t_i^- and the offset, $\Delta x_i^-, \Delta y_i^-$, producing the best match:

$$\begin{aligned} x_i^- &= s_i^- + \Delta x_i^- & \Delta x_i^- &\in \{-1,0,1\} \\ y_i^- &= t_i^- + \Delta y_i^- & \Delta y_i^- &\in \{-1,0,1\} \end{aligned} \tag{5}$$

At the top level (n) of the motion estimation pyramid (where the motion estimation begins) the coordinates of the starting block are the same as those of the block being matched (as in level *i* of Figure 5):

$$\begin{aligned} s_n^- &= x_n \\ t_n^- &= y_n \end{aligned} \tag{6}$$

Below the top level of the pyramid the coordinates of the starting block are given by the following equations:

$$\begin{aligned} s_{i-1}^- &= x_{i-1} + 2x_i^- - \Delta x_i^- & 2 \le i \le n \\ t_{i-1}^- &= y_{i-1} + 2y_i^- - \Delta y_i^- & 2 \le i \le n \end{aligned} \tag{7}$$

where x_{i-1}, y_{i-1} are the coordinates of one of the four blocks which are daughters of the block with coordinates x_i, y_i.

The matching process can now be described precisely:

(1) For each block at level n in frame *j*, set the starting block for the search according to equation (6); then find the optimal values of the offset $\Delta x_n^-, \Delta y_n^-$; then substitute into equation (5) to obtain the coordinates x_n^-, y_n^- of the best matching block.

(2) For each block in the next lower level of frame *j*, use equation (7) to obtain the coordinates of the starting block; then find the optimal values of the offset; then substitute into equation (5) to obtain the coordinates of the best matching block.

(3) Repeat step (2) until all levels have be processed.

To obtain more precise motion estimates, we also provide the option of a second round of block matching using a ± 0.5 element step size. This second round of matching is carried out in the neighborhood of the block that gave the best match using the ± 1 element step size.

Motion compensation is performed using the motion estimates from the block matching procedure. Specifically, for each level of the pyramid, each block of pyramid coefficients in the current frame is subtracted (element by element) from the best matching block of the previous frame. To reduce the effects of motion estimation errors, the zero motion vector is also tested, and then selected if it provides better compensation.

6. BAND-PASS MULTIRESOLUTION CODING

Each of the motion-compensated foveation regions (see Figure 3) is coded using a multiresolution transformation (e.g., Laplacian pyramid, wavelet pyramid, discrete cosine transform). For the examples presented here, we used the Laplacian pyramid because of its good real-time performance. However, with moderate degrees of foveation, the foveation regions are small enough that useable real-time performance should be obtained with wavelet pyamids or with the discrete cosine transform.

7. THRESHOLDING AND QUANTIZATION

With foveated multiresolution pyramids, it is possible to obtain compression, with minimal loss of perceptual quality, by thresholding the transform coefficients on the basis of psychophysical data measured as a function of both the spatial frequency and the eccentricity. The thresholding function we use is essentially the same as equation (1):

$$\delta = T_{\max} CT_1 \exp\left(\alpha f \frac{e + e_2}{e_2} \right) \tag{8}$$

where δ is the value of the threshold, $T_{\max}$ is the maximum absolute value of the transform coefficients, and the remaining constants and variables are the same as in equation (1). If a transform coefficient falls below the threshold then its value is set to zero:

$$if \quad |T(x,y)| \leq \delta \quad then \quad T(x,y) = 0 \tag{9}$$

In applying equation (8), we allow the minimum contrast threshold parameter, CT_1, to be different from the value, CT_0, used to determine the sizes of the foveation regions.

To obtain further compression, we quantize the significant (non-zero) transform coefficients. In general, the higher resolution levels of the transformation can be quantized to a greater degree than the lower resolution levels, without objectionable loss of image quality. Therefore, the number of quantization levels is set to a minimum value, $NQ_{\min}$, for the highest resolution coefficients, and is increased logarithmically, reaching a maximum value, $NQ_{\max}$, for the lowest resolution coeffients. With the Laplacian pyramid, we obtain better image quality with nonuniform quantization than with uniform quantization. Specifically, for each level of the pyramid, we bin the significant coefficients so that each bin contains approximately the same number of coefficients; the quantization value for each bin is taken to be the mean of the coefficient values in the bin.

8. ZERO-TREE AND ARITHMETIC CODING

Two forms of lossless coding are performed before data transmission. The first is a simple two-pass form of zero-tree coding. The first pass scans each level of the pyramid, in non-overlapping 2x2 blocks, beginning at the lowest level of the pyramid (the highest resolution). If all four elements in a block are zero, or are "zero root" symbols, then the parent element in the next higher level of the pyramid is checked; if the parent is also zero then the four elements are replaced with a "null" symbol indicating that they are not to be transmitted, and the parent is replaced with a zero root symbol. This process continues to the highest level of the pyramid. The second pass scans (in a fixed known order) all the elements starting at the highest level of the pyramid; all symbols except the null symbols are entered into the output data stream. Because, the scanning is in a fixed order, all of the zeros "under" a zero root symbol can be placed in their correct locations during reconstruction. The second and final lossless coding step is standard arithmetic coding.[19]

9. RECONSTRUCTION

Reconstruction proceeds by inversing the coding stages in the reverse order that they were applied.

Figure 6. Foveated images (680 x 768) of a macaque monkey. The foveation point is indicated by the small plus/cross. A. Strong foveation resulting in a factor of 19 reduction in the number of pyramid elements (CT_0 = 0.25, $\alpha = 0.1$, $e_2 = 2.3$, w = 10, r_0 = 2, deg/pixel = .046). B. Moderate foveation resulting in a factor of 5.5 reduction in the number of pyramid elements (CT_0 = 0.05, $\alpha = 0.1$, $e_2 = 2.3$, w = 10, r_0 = 2, deg/pixel = .046).

10. SYSTEM PERFORMANCE

Our first real-time implementation demonstrates the following components of the full system: pointing device input, foveated low-pass pyramid coding, blending and interpolation, Laplacian pyramid coding, decoding, displaying. Figure 6 shows two example output images (680 x 768, 8-bit gray scale), obtained with a 3 x 3 kernel (for both the low-pass pyramid and the Laplacian pyramid),

$$K = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix} \qquad (10)$$

and a blending function width of 10 elements. The image on the left has been strongly foveated (factor of 19 reduction in the number of elements), and the one on the right has been moderately foveated (factor of 5.5 reduction). The small crosses indicate the foveation point. On a single 300 MHz Pentium Pro, 800 x 600 images are processed through all five components above at approximately 25 frames per second for the strong foveation, and approximately 20 frames per second for the moderate foveation. The frame rate is 50% higher for 640 x 480 images. Furthermore, these numbers underestimate performance for many applications (e.g., surveillance and teleoperation), because the coding, blending and interpolation would be done on a processor at the remote site, while the decoding and displaying would be done on another processor at the control site (see Figure 1). The first three components (pointing device input, foveated low-pass pyramid coding, blending and interpolation) could serve as a software preprocessor for a hardware MPEG coder.

Our second real-time implementation demonstrates all of the components in the full system. Although yet not fully optimized for real-time perfomance, we have obtained some preliminary results for three different video sequences: "Claire", "Mobile and Calender" and "Mall." The uncompressed entropy for of "Claire" is 6.3 bits/pixel. Figure 7A shows frame 22 of the uncompressed sequence. Figure 7B shows frame 22 of the compressed sequence (0.043 bits/pixel for I frame plus P frames, PSNR = 36.7 dB). Figure 7C shows frame 22 of the compressed and foveated sequence (0.020 bits/pixel). Foveation only adds a little to the total compression because the motion is primarily confined to the unfoveated region and because the image is small (360 x 288). The "Mobile and Calender" sequence better demonstrates of the value of foveation. Figure 8A shows frame 10 of the compressed sequence (0.08 bits/pixel for P frames, 28.1 dB). Figure 8B shows frame 10 of the compressed and foveated sequence with reduced thresholding and quantization so that the compression remains approximately the same (0.08 bits/pixel, 31.6 dB in the foveation region). This example demonstrates how foveation can be traded with quantization to dynamically allocate resolution to points of interest, without increasing bandwidth requirements (notice the greatly reduced number of artifacts in B near the foveation point on the ball). These 512 x 400 images have been

Figure 7. "Claire." A. Uncompressed (6.3 b/pix) B. Compressed (0.043 b/pix, 36.7 dB) C. Foveated (0.020 b/pix)

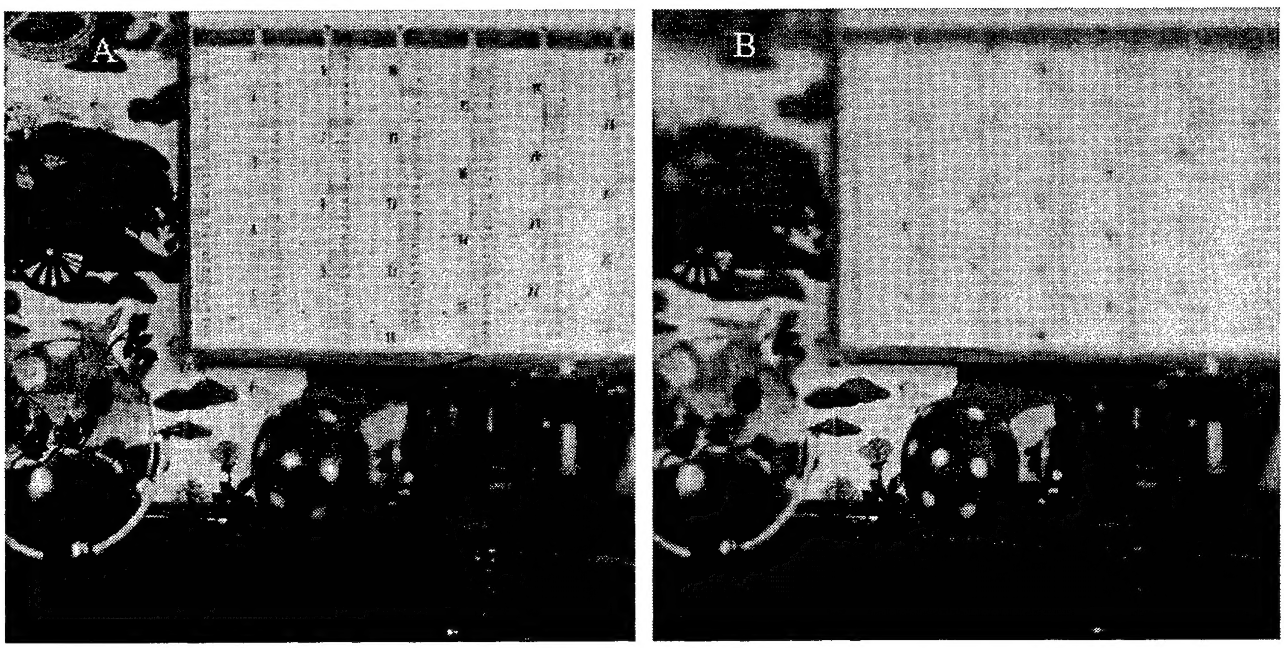


Figure 8. "Mobile and Calender." A. Compressed (0.08 b/pix, 28.1 dB) B. Foveated (0.08 b/pix, 31.6 dB in fovea)

clipped on the left to make the quantization artifacts in Figure 8A more visible in the reproduced images. The "Mall" sequence illustrates the value of foveation in applications such as surveillance or teleoperation. Figure 9 shows frame 10 of the compressed and foveated sequence (0.013 bits/pixel for the P frames, 29.3 dB in the foveated region). The unfoveated compressed sequence is not shown (0.092 bits/pixel, 29.3 dB). Foveation increased the compression by a factor of 7.

We cannot yet report the speed performance of the full system because there are several inefficient steps with obvious remedies. Nonetheless, as it currently stands the encoding rate for foveated "Claire" (Figure 7C) is 19 frames/sec and the decoding rate is 94 frames/sec.

11. APPLICATIONS

One of the more obvious applications of foveated imaging is in teleoperation, where there is often motion over extended image regions, and where bandwidth limitations are usually severe due to the need for wireless communication. For example, during teleoperation of a vehicle, high resolution information is required primarily in the heading direction for path planning and for avoidance of obstacles and hazards; but, relatively low resolution is sufficient in the periphery for judgments

of heading from optical flow, and for detection of incoming objects and/or vehicles. With the wide field of view usually desired for teleoperation, foveated imaging can have a truly dramatic affect on bandwidth transmission requirements.[8]

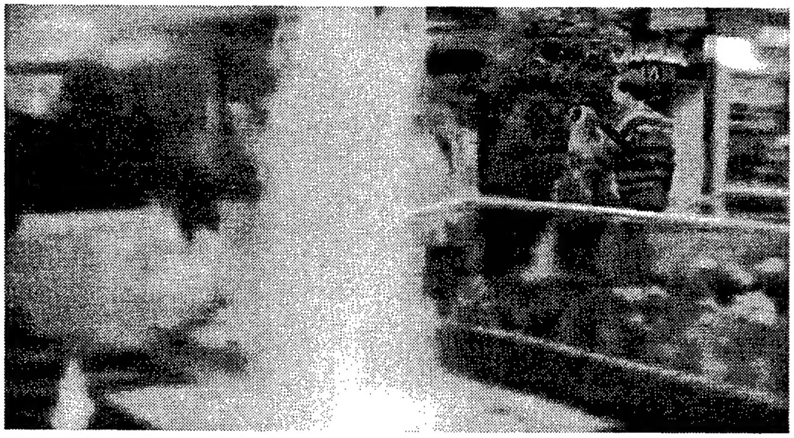


Figure 9. "Mall" Compressed--not shown (.092 b/pix, 29.3 dB), Foveated (0.013 b/pix),

Other potential applications for foveated imaging are in surveillance, telemedicine and teleconferencing. In these applications, there are often localized regions of the video images that the user wants to inspect. Foveated imaging allows the user to dynamically allocate high spatial resolution to the regions of interest (see Figure 8).

Obviously, to make optimal use of foveated imaging, the resolution of the video information to be transmitted (e.g., the camera resolution) must be selected or created to exceed the bandwidth limitations of the communication channel when using the non-foveated codec at the desired image frame rate. Foveated imaging will then allow the user to access the high resolution video information that could not be accessed (at the desired frame rate) without foveated imaging.

The most elegant and seamless implementation of foveated imaging is with an eyetracker, which keeps the high-resolution region of the displayed image centered on the observer's line of sight. For example, Owl Displays Inc. (Austin, TX) is currently integrating the FMP imaging system into an elegant high resolution helmet mounted display system with a built in eyetracker. For moderate degrees of foveation in this system, the user cannot detect that the images are foveated.

On the other hand, foveation is often valuable even using simpler, less expensive and more robust pointing devices, such as a mouse or touch pad. For example, in teleoperation, directing the foveation point toward the heading direction will provide fine detail where it is most needed, but at the same time, provide a wide field of view. Although the foveation will sometimes be visible, the user will perform better than without foveation, given a fixed communication bandwidth. Similarly, in surveillance, telemedicine and teleconferencing it is often sufficient, for getting a particular task done, to direct the foveation point to regions of interest with a simple pointing device.

One way to think about value of foveated imaging is to consider being confronted with the choice of two nearly equivalent video communications systems. Both transmit information at the same bandwidth with equal resolution in a non-foveated mode, but one system gives the user the option of switching to a foveated mode where spatial resolution can be dynamically allocated to regions of interest without affecting frame rate. A little consideration leads one to the conclusion that there are many situations where this feature would very valuable in allowing the user to complete a task that would be difficult or impossible otherwise. This feature would be valuable even though the image degradation outside the foveation region might be visible, as it would be with strong foveation or with pointing devices other than an eyetracker.

12. CONCLUSION

This paper describes a foveated multiresolution pyramid (FMP) coder/decoder for low bandwidth video communications. The codec, although not yet fully honed, provides smooth foveation and good compression at useful frame rates on a general purpose computer (a Pentium running under Windows95/NT). The novel contributions include: (1) full integration of foveation into multiresolution pyramids, (2) the development of efficient pyramid, foveation, and motion estimation algorithms which make possible real-time operation on conventional computer hardware, (3) development of efficient methods for eliminating foveation artifacts, (4) the use of psychophysical contrast sensitivity data as function of spatial frequency and

eccentricity to determine foveation regions, and to determine the thresholding and quantization. Our experience suggests that foveated imaging would be a useful feature in many video communications applications.

13. ACKNOWLEDGMENTS

This research was supported by AFOSR STTR grant F49620-94-C-0090 to WSG and to OWL Displays Inc., Austin TX, and by AFOSR URI grant F49620-93-1-0307. Larry Stern and Carl Creeger provided valuable technical assistance.

14. REFERENCES

1. C. M. Howard, "Display Characteristics of Example Light-Valve Projectors," Operations Training Division, Air Force Human Resources Laboratory, Williams AFB, AZ AFHRL-TP-88-44, 1989.
2. R. D. Juday and T. E. Fisher, "Geometric transformations for video compression and human teleoperator display," *SPIE Proceedings: Optical Pattern Recognition*, vol. 1053, pp. 116-123, 1989.
3. C. F. R. Weiman, "Video Compression Via Log Polar Mapping," *SPIE Proceedings : Real Time Image Processing II*, vol. 1295, pp. 266-277, 1990.
4. B. B. Benderson, R. S. Wallace, and E. L. Schwartz, "A miniature pan-tilt actuator: the spherical pointing motor," *IEEE Transactions Robotics and Automation*, vol. 10, pp. 298-308, 1994.
5. H. D. Warner, G. L. Serfoss, and D. C. Hubbard, "Effects of Area-of-Interest Display Characteristics on Visual Search Performance and Head Movements in Simulated Low-Level Flight," Armstrong Laboratory, Human Resources Directorate, Aircrew Training Division, Williams AFB, AZ. AL-TR-1993-0023, 1993.
6. P. L. Silsbee, A. C. Bovik, and D. Chen, "Visual pattern image sequence coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, pp. 291-301, 1993.
7. B. S. Barnett and A. C. Bovik, "Motion compensated visual pattern image sequence coding for full motion multisession videoconferencing on multimedia workstation," *Journal of Electronic Imaging*, vol. 5, pp. 129-143, 1996.
8. P. T. Kortum and W. S. Geisler, "Implementation of a foveated image-coding system for bandwidth reduction of video images," *SPIE Proceedings: Human Vision and Electronic Imaging*, vol. 2657, pp. 350-360, 1996.
9. E. Chang and C. K. Yap, "A wavelet approach to foveating images," *ACM Symposium on Computational Geometry*, vol. 13, pp. 397-399, 1997.
10. P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. COM-31, pp. 532-540, 1983.
11. A. M. Tekalp, *Digital Video Processing*. Upper Saddle River: Prentice Hall, 1995.
12. E. H. Adelson, E. Simoncelli, and R. Hingorani, "Othogonal pyramid transforms for image coding," *SPIE Proceedings: Visual Communications and Image Processing II*, vol. 845, pp. 50-58, 1987.
13. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavlet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205-220, 1992.
14. S. A. Martucci, I. Sodogar, T. Chiang, and Y. Zhang, "A zerotree wavelet video coder," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 109-118, 1997.
15. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445-3462, 1993.
16. J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field.," *Vision Research*, vol. 21, pp. 409-418, 1981.
17. T. L. Arnow and W. S. Geisler, "Visual detection following retinal damage: Predictions of an inhomgeneous retino-cortical model," *SPIE Proceedings: Human Vision and Electronic Imaging*, vol. **2674**, pp. 119-130, 1996.
18. M. S. Banks, A. B. Sekuler, and S. J. Anderson, "Peripheral spatial vision: limits imposed by optics, photoreceptors, and receptor pooling," *Journal of the Optical Society of America*, vol. 8, pp. 1775-1787, 1991.
19. I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Communications of the ACM*, vol. 30, pp. 520-540, 1987.

Further author information -

W.S.G. (correspondence): Email: geisler@psy.utexas.edu; Telephone: 512-471-5380; Fax: 512-471-7356

J.S.P.: Email: jsp@mail.utexas.edu; Telephone: 512-471-3054; Fax: 512-471-7356

Invited Presentation: Society for Information Display (SID), San Jose, June 1999.

Variable resolution displays for visual communication and simulation

Wilson S. Geisler and Jeffrey S. Perry

Center for Vision and Image Sciences, University of Texas at Austin, Austin TX 78712

The spatial resolution of the human visual system declines precipitously away from the point of gaze, and thus it is possible to decrease gradually the display resolution in the visual periphery with little effect on perceptual quality or visual performance. This fact can be exploited in various applications to increase image compression, to increase image processing speed, and to decrease access time for image data. This paper describes a multiresolution pyramid method for creating variable resolution displays in real time using general purpose computers (e.g., a Pentium with the Windows 95/98/NT OS). The location of the high resolution region(s) can be dynamically controlled by the user with a pointing device (e.g., a mouse or an eye tracker) or by an algorithm. Our method has a number of advantages: high computational speed and efficiency, smooth artifact-free variable resolution, and compatibility with other image processing software/hardware. The real-time software will be demonstrated and a number of potential applications will be described, including variable resolution MPEG, variable resolution image retrieval, and variable resolution 3D simulation. Some of the real-time software demonstrations are currently available at http://fi.cvis.psy.utexas.edu.

Support: AFOSR grants F49620-94-C-0090 and F49620-93-1-0307.